|  |  |
| --- | --- |
| **Quiz No. 1 Skill Test** | |
| **Course Code:** CPE 201L | **Program:** BSCpE |
| **Course Title:** Data Structures and Algorithms | **Date Performed:** August 30, 2025 |
| **Section:** 2 - A | **Date Submitted:** August 30, 2025 |
| **Name:** Hermosura, Leigh B. | **Instructor:** Ma’am Maria Rizette H. Sayo |
| 1. **Objectives** | |
| * Choose one type of Data Structure (Array, Linked-List (Singly, Doubly), Stack, Queue) * Create a Python program that appends each character of your full name * Traverse each character | |
| **2. Discussion** | |
| Data Structure is a specialized format of organizing, storing, and managing data, allowing it be accessed and manipulated efficiently.  Data Structures can be classified into two, Linear and Non-Linear. Non-Linear Data Structures include Trees, Graphs, Tables, and Sets while Linear Data Structures include Arrays, Linked-Lists, Stacks, and Queues.  Linear Data Structures are a type of data structure where data elements are arranged sequentially. Each element has a previous and next adjacent with the exemption of the first and last elements. | |
| **3. Materials and Equipment** | |
| The materials and equipment used in this activity are the following:   * **Computer** – used to be able to write, debug, and run code. * **Google Colab** – a cloud-based service to run and write python code in a Jupyter Notebook environment. | |
| **4. Procedure** | |
| The Data Structure that I chose in this activity is a Queue under Linear Data Structure where the data follows a First-in First-out (FIFO) procedure of enqueuing and dequeuing items into the queue.  In the program, I first started by creating a *Queue* class then initializing an empty queue. Next, I defined *enqueue* and *dequeue* functions to be able to add and remove items from the queue. The function *printQueue* prints *self.queue* and displays the list of items. The *isQueueEmpty* function checks whether the queue is empty or not. The function *queueLength* gets the length of the number of items inside the queue and prints its value. Lastly, the *firstIn* function displays the first item appended in the queue.  Outside the *Queue* class is the variable called *myName* which stores the string “Leigh Buena Hermosura”. The class *Queue* is then set to the variable *MyQueue* to access the class’s functions. First, the function *isQueueEmpty* is used to check if *MyQueue* is empty, which returns that it is empty. The function *printQueue* is called to display the empty queue and *queueLength* to check the length of the list. The *enqueue* function is called for *MyQueue* and it passes the variable *myName* to be appended in the queue. The function *enqueue* iterates through each character in the string and appends them to the queue. The program then checks if the queue is empty, the length of the queue, and the first item to be appended in the queue. | |
| **5. Output** | |
|  | |
| **7. Conclusion** | |
| This activity demonstrates what I have learned about Data Structures. Queue Data Structure uses the First-in First-Out (FIFO) procedure. The functions *enqueue*, *dequeue*, *isQueueEmpty*, *queueLength*, and *firstIn* shows the process of adding, removing, and checking the length, and getting the first item in the queue. It highlights the difference between each type of Data Structure and how to utilize them in various applications. | |